Introdução aos Testes Automatizados

Paulo Cheque (paulocheque@agilcoop.org.br)

Cursos de Verão 2010

Sobre

- Linguagem
- Tipos de Testes
- Introdução aos Arcabouços de Teste

Muitas comunidades

- Linguajar distinto:
  - Comunidades Ágeis
  - Comunidades Formais
  - Comunidade de QA
  - Ferramentas
  - Clientes

Padrões

- - BS 7925-2:1998. Software Component Testing.
- - DO-178B:1992. Software Considerations in Airborne Systems and Equipment
- Certification, Requirements and Technical Concepts for Aviation (RTCA SC167).
- - IEEE 610.12:1990. Standard Glossary of Software Engineering Terminology.
- - IEEE 829:1998. Standard for Software Test Documentation.
- - IEEE 1008:1993. Standard for Software Unit Testing.
- - IEEE 1012:1986. Standard for Verification and Validation Plans
- ...

Jargões

- Defeito/Engano/Erro/Falha
- Verificação
- Validação
- Inspeção
- Depuração
- ...

Termos

- Caixa-Preta/Branca/Cinza/Vidro/Transparente
- Teste Funcional
- Teste Estrutural

- Teste
- Casos de Teste
- Bateria de Testes

Siglas

- SUT/AUT: System/Application Under Test
- TFD/POUT: Test-First Development / Plain Old Unit Test
- TAD: Test-After Development
- TDD: Test-Driven Development / Design
- BDD/EDD/STDD: Behaviour/Example/Story Test Driven Development
- TDDD: Test-Driven Design Databases

Dublês

- ***Dummy***: Objeto utilizado apenas para permitir a criação/execução do teste
- ***Fake***: Objeto leve que contém uma implementação falsa
- ***Stub***: Objeto contendo informações que serão utilizadas pelos testes
- ***Mock***: Objeto que possui uma interface apropriada e que contém informações e comportamentos que serão utilizados pelos testes
- ***Spy***: Objeto que captura chamadas indiretas

Tipos de Testes

- Unidade/Integração
  - Persistência de dados
  - Objetos
  - Aspectos
- Interface
  - Unidade
  - Integração
- Aceitação
  - Usuário / Cliente
- Desempenho
- Carga / Estresse
- Longeividade
- Segurança
- Layout / Beleza
- Usabilidade
- Acessibilidade
- ...

Teste de Unidade

- Unidade: Classe/Módulo ou Método/Função
- Unidade ou “mini-integração”

- Teste do código fonte com foco na funcionalidade
- Teste básico e muito importante
- Principal teste para verificar correção
- Teste muito específico
  - Evita desperdício de tempo com depuração

Os Primeiros Testes de Unidade

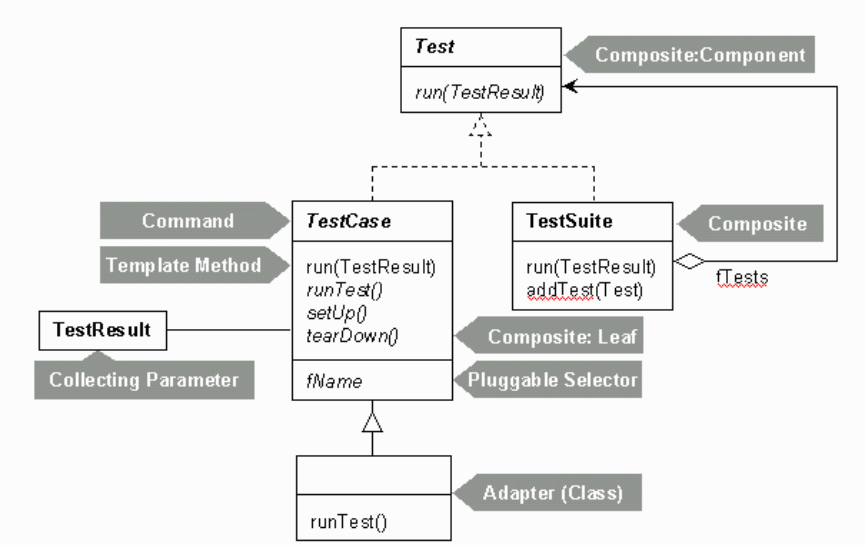
- Main
- Arcabouços x-Unit
  - Casos de Teste
  - Verificações
  - Exceções
  - Resultados
  - Relatórios

Testes automatizados sem uso de ferramentas

```
public class MainExample {

    public static void main(String[] args) {
        if(!new Double(Math.sqrt(-1)).isNaN())

            throw new RuntimeException("Ops, falhou!");

        // Ou

        assert new Double(Math.sqrt(-1)).isNaN() == true;

    }
}
```

xUnit

- Família de ferramentas para testes de unidade:
  - JUnit (Java), TestNG (Java), Nunit (.NET), CSUnit (C#), DUnit (Delphi), VBUnit (Visual Basic) ...
- Ótimo artigo para aprender mais sobre arcabouços xUnit e sobre orientação a objetos:



- http://junit.sourceforge.net/doc/cookstour/cookstour.htm

Junit 4.x: Casos de Teste

- Convenções

```
public void testMetodoDeTeste() { ... }
```

- Anotações

```
@Test public void metodoDeTeste() { ... }
```

- Configurações
  - Arquivos XML

JUnit 4.x: Verificação

- asserttrue / assertfalse
- assertEquals
- assertSame
- fail
- Expect Exceptions

- Mensagens amigáveis
- Hamcrest: AssertThat

JUnit 4.x: Exemplo

```
public class RaizTest {
    @Test public void raizDe4Eh2() {
        assertEquals(2, Math.sqrt(4));
    }

    @Test public void raizDeNegativoNaoExiste() {
        assertTrue(new Double(Math.sqrt(-1)).isNaN());
    }
}
```

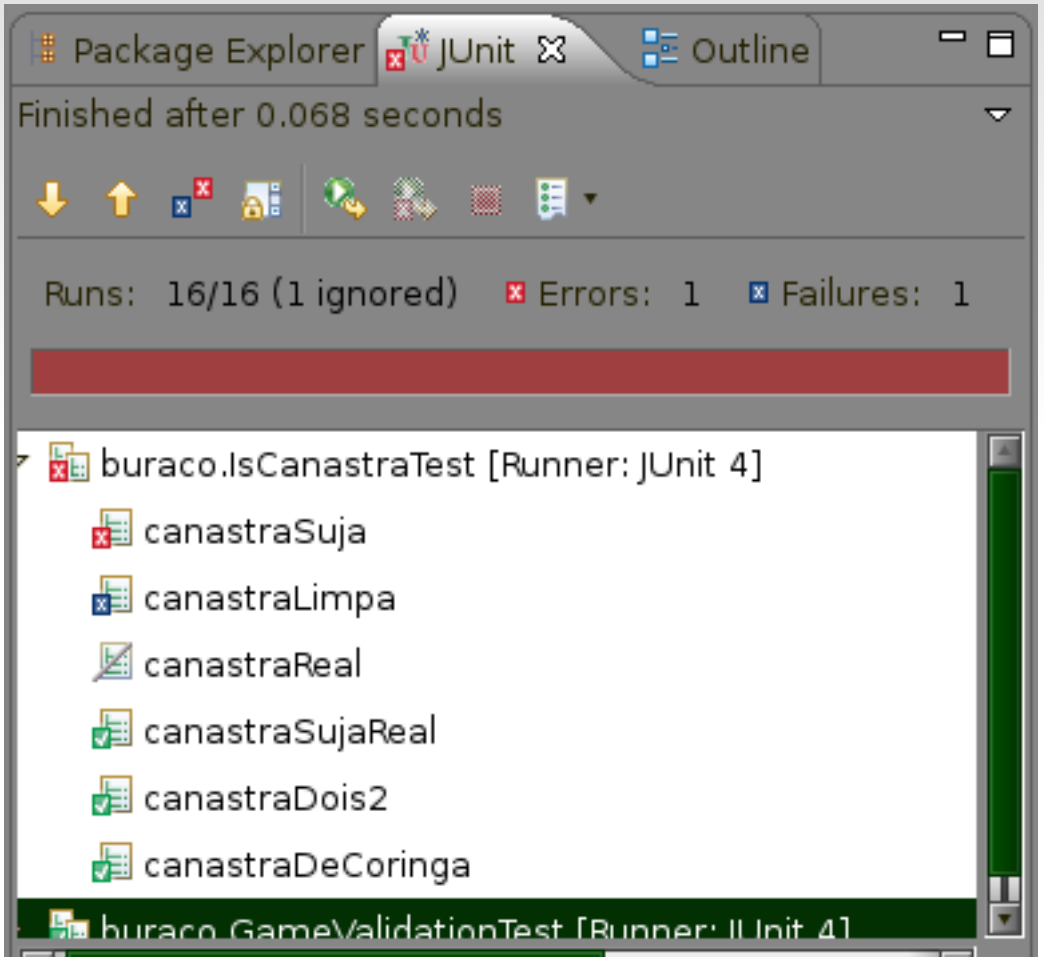
Exemplo: Folha de Pagamento

```
public class CalculaDoSalarioTest {

    @Test public void remuneracaoDeEstagiarioTemImpostoReduzido() {
        SimpleDateFormat formatador = new SimpleDateFormat("dd/MM/yyyy");
        Date dataInicioDoPeriodo = formatador.parse("01/08/2009");
        Date dataFimDoPeriodo = formatador.parse("01/09/2009");
        FolhaDePagamento estagiario = new FolhaDePagamentoEstagiario(536);
        int salario = estagiario.salario(dataInicioDoPeriodo, dataFimDoPeriodo);
        assertEquals(499, salario));
    }

    @Test public void remuneracaoDeContribuinte() { … }
    @Test public void impostoDeXPorcentoParaTrabalhadorAvulso() { … }
    @Test public void impostoDeXPorcentoParaEmpregadoSegurado() { … }
}
```

Junit 4.x: Resultados

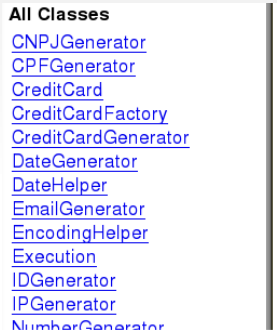
- Testes ignorados
- Falhas
- Erros
- Sucesso

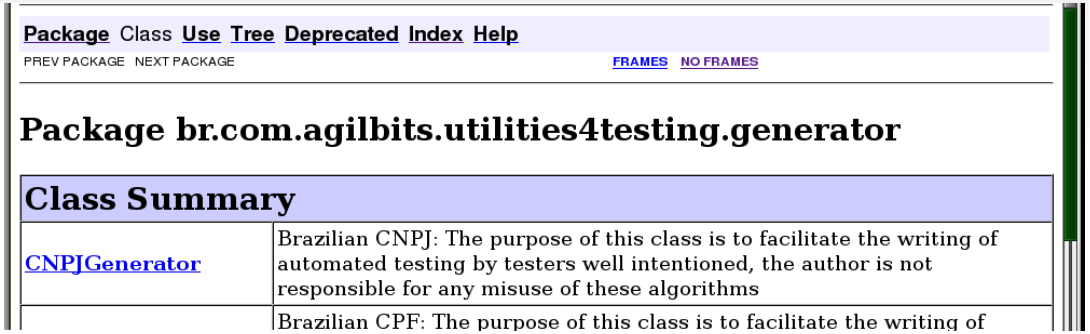
- Mensagens de erros
- Pilha de execução
- Linha do erro
- ...

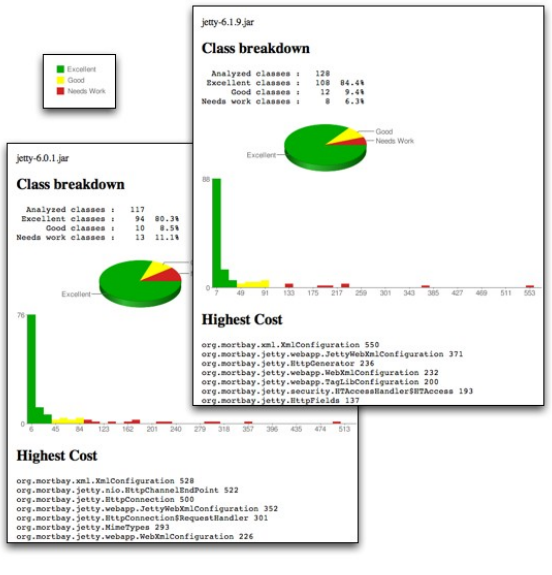


Relatórios de ferramentas de teste

- Linguagem
  - API-Docs das classes de testes
- Ferramentas
  - Logs
  - Screenshots
- Métricas
  - Cobertura, testabilidade ...







Teste de Integração

- Em geral: Teste que envolve mais de uma camada ou que trabalha com todas as dependências
  - Persistência
  - Outros sistemas / *Web Services*

- Verifica erros da integração entre módulos, componentes e camadas que podem funcionar perfeitamente individualmente

Teste de Interface de Usuário

- Console, GUI, WUI

- Camada crítica a erros
  - Exposta aos usuários: infinitas combinações de uso
- Útil também para:
  - Testes de aceitação
  - Testes de usabilidade
  - Testes de layout

Testes de Aceitação

- Teste do Cliente
- Testa as funcionalidades do ponto de vista do cliente
- Garantia que o sistema faz o esperado
- Clientes/Usuários e desenvolvedores podem trabalhar em parceria na escrita dos testes
  - Detalhes de implementação ocultos
  - Linguagem do cliente
- Ferramentas: Fitnesse, Fit, Selenium ...

Testes de Desempenho

- Avaliar tempo de resposta de um módulo específico
- Profilers ajudam a encontrar gargalos

- Não avaliam a complexidade computacional
- Resultados dependem da infra-estrutura de hardware: rede, processador, memória ...

Teste de Carga

- Simula grandes quantidades de:
  - Usuários, Requisições, Dados ...
- Requisições: Simultâneas ou dentro de um intervalo de tempo
- Testa:
  - Infra-estrutura
  - Hardware
  - Banda de rede
  - Banco de dados e servidores
- Ferramentas: JMeter ...

Teste de Estresse

- Teste de Carga Máxima
- Identificar quantidade máxima de:
  - Usuários
  - Requisições
  - Dados
- Encontrar limitações do ambiente

- Ferramentas: JMeter, Jstress ...

Teste de Longeividade

- Teste que verifica o desempenho e a correção de um sistema dentro de um grande período de tempo de execução
- Encontrar problemas com vazamento de memória
- Encontrar erros de cache

- Ferramentas: JMeter

Teste de Segurança

- Útil para aplicações expostas a usuários mal-intencionados
  - Web, Caixas eletrônicos ...
- Testa:
  - Falhas de permissão na interface
  - Falhas no modelo (SQLInjection ...)
  - Bugs nos produtos de servidores
  - Fragilidade de ataques DoS (Teste de estresse)

Informações Pertinentes

- **Máquina**: CPU, Memória, HD, Swap ...
- **SO**: Nome/Distribuição, Versão, Kernel, Locale ...
- **Servidores**: Nome-Versão, Arquivos Pertinentes
  - Tomcat 5.5, Jboss 4.0.4, (/etc/init.d/jboss)
- **Bancos de Dados**: Nome-Versão, Máquinas, Schema
- **Rede**: IP, Firewall, Portas, DNS ...
- **Dependências**: Nome-Versão
  - JDK 1.5.0_06, Ruby 1.8
- **Produto**: Nome, Versão, data instalação, logs

Teste Fumaça

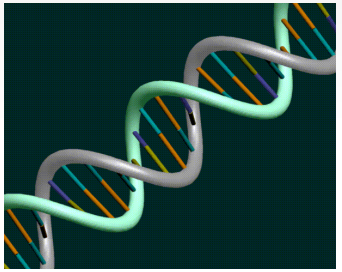
- Testes Rápidos, Superficiais e Abrangentes
- Realizados após a instalação do software
- Exemplo:
  - Acesse uma página Web e verifique que não apareceu o texto “404”

Teste de Sanidade

- Um Teste Fumaça
- Apropriado para buscar erros absurdos em algoritmos
- Utilizar técnicas relacionadas ao algoritmo para identificar possíveis falhas
  - Teoremas
  - Propriedades matemáticas

Teste de Mutação

- Teste dos Testes
- *"Testes podem mostrar a presença de erros, não a sua ausência"* - Dijkstra
- Infinitas possibilidades de erros?
  - Programador competente
- Mutante: SUT com pequenas alterações
- Executar os testes sob os Mutantes
- Análise dos Resultados: Mutantes e Testes



Outros tipos...

- Teste de:
  - Instalação: Verifica se todos os componentes se interconectaram e se não houve incompatibilidades com o hardware
  - Recuperação: Reação adequada após erros
  - Configuração: Configurações diferentes e ambientes distintos (portabilidade)
  - ...
- Versões Alfa / Beta

Algumas Ferramentas

- Testes de Unidade:
  - CxxTest (C++): http://cxxtest.sourceforge.net
  - CUnit: http://cunit.sourceforge.net
  - JUnit (Java): http://www.junit.org
  - DUnit (Delphi): http://dunit.sourceforge.net
  - VBUnit (Visual Basic): http://www.vbunit.com
  - TestNG (Java): http://testng.org
  - RSpec (Ruby): http://rspec.info/
- http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks

+Algumas Ferramentas

- Mock Objects:
  - SevenMock (Java): http://seven-mock.sourceforge.net
  - EasyMock (Java): http://www.easymock.org/
  - JMock (Java): http://www.jmock.org
  - Rhino.Mocks (.NET): http://www.ayende.com/projects/rhino-mocks.aspx
  - SMock (Smalltalk): http://www.macta.f2s.com/Thoughts/smock.html
  - Mockpp (C++): http://mockpp.sourceforge.net
  - GoogleMock (C++): http://code.google.com/p/googlemock

+Algumas Ferramentas

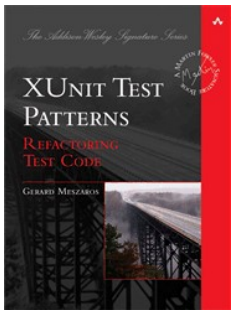
- Testes de Mutação:
  - Jabuti (USP São Carlos): http://jabuti.incubadora.fapesp.br
  - Jester: http://jester.sourceforge.net
  - Heckle: https://rubyforge.org/projects/seattlerb
- Testes de Desempenho / Carga / Estresse / Longeividade:
  - JMeter: http://jakarta.apache.org/jmeter
- Outros:
  - JPDFUnit: http://jpdfunit.sourceforge.net

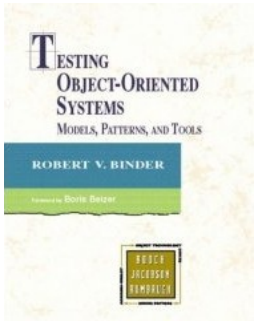
Alguns links

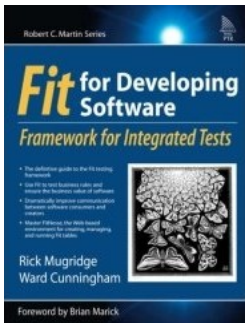
- http://www.testing.com
- http://www.opensourcetesting.org
- http://xunitpatterns.com
- http://www.mockobjects.com
- http://java-source.net/open-source/testing-tools
- http://www.junit.org
- http://www.agilcoop.org.br

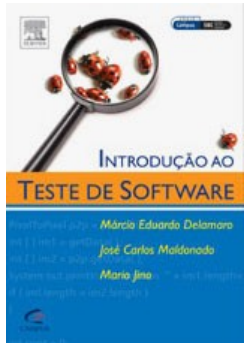
Alguns Livros

- Gerard Meszaros, “xUnit Test Patterns, Refactoring Test Code”, Addison-Wesley, 2007
- Robert V. Binder, “Testing Object-Oriented Systems”, Addison-Wesley Professional, 1999
- L. Crispin, T. House, “Testing Extreme Programming”, Addison-Wesley, 2005
- R. Mugridge, W. Cunningham, “Fit for Developing Software”, Prentice Hall, 2006
- M. Delamaro, J. Maldonado, M. Jino, “Introdução ao Teste de Software”, Campus, 2007









Contato

http://www.agilcoop.org.br

http://ccsl.ime.usp.br

paulocheque@agilcoop.org.br